

Normas

- El examen puntúa sobre **8 puntos**.
- La duración total del mismo es de **35 minutos**.
- Se deberá tener el DNI o el carnet de la UPM en lugar visible.

Cuestionario

- (1 punto) 1. Dada la siguiente declaración, qué salida producen las instrucciones printf

```
char str1[80] = "cinco";
printf("1.Long_str1:_%d\n", (int) strlen(str1));
printf("2.str1[%d]:_%c\n", (int) strlen(str1)-1, str1[strlen(str1)-1]);
printf("3.str1[5]:_%c\n", str1[5]);
```

Solución:

1.Long str1: 5
2.str1[4]: o
3.str1[5]:

- (1 punto) 2. Dada la siguiente declaración de la variable vector

```
int vector[5]={1,2,3,4,5}
```

Implementar un bucle que sume los elementos del vector.

Solución:

```
suma=0;
for (i=0;i<5;i++) {
    suma=suma+vector[i];
}
printf("SUMA_%d\n", suma);
```

- (1 punto) 3. Escriba la salida que generaría la ejecución del siguiente programa en C :

```
#include <stdio.h>
int main (int argc, char*argv[])
{
    int lista[5]= {1,2,3,4,5};
    int *p;
    p=lista;
    printf("*P=_%d\n", *(p+1));
    printf("*P=_%d\n", p[1]);
}
```

```
p=&lista[1];  
printf("P=_ %d\n", *p);  
  
return 0;  
}
```

Solución:

```
*P = 2  
*P = 2  
*P = 2
```

- (1 punto) 4. Si al ejecutar un programa obtenemos como resultado Segmentation fault (core dumped) y no se genera ningún fichero core. A qué se puede deber? Qué comando permitiría que se generara el fichero core correspondiente?

Solución: El shell tiene limitado el tamaño de los ficheros core a 0.
ulimit -c unlimited

- (1 punto) 5. Indique la función de librería estándar en C para abrir un fichero:

Solución:

fopen

- (1 punto) 6. En un programa en C, indique cómo deben ser los parámetros de la función 'main' cuando, en dicha función, se quiere acceder a los argumentos proporcionados en línea de comandos.

```
int main( <<PARTE_A_INDICAR EN LA RESPUESTA>> ) {  
  
}
```

Solución:

```
int argc, char *argv[]  
// o bien  
int argc, char **argv
```

- (1 punto) 7. Sea el siguiente extracto de código en lenguaje C:

Apellidos:

Nombre:

Matrícula:

```
#include <stdlib.h>
#include <stdlib.h>

int main() {
    int n;
    int *ptr;

    ptr = &n;
    /* ... */
    free( ptr );
    return 0;
}
```

Indique si dicho código es correcto (responda SÍ/NO.)

Solución: NO

- (1 punto) 8. Cómo podríamos saber cuanta memoria consume una variable de tipo puntero a un entero? Mostrarlo con un ejemplo

Solución:

```
int *p;
printf("Memoria_consumida:_%d_bytes", sizeof(p));
```